Urban Dictionary Embeddings for Slang NLP Applications

Urban Dictionary Embeddings for Slang NLP Applications

Steven R. Wilson[1], **Walid Magdy**[1,2], **Barbara McGillivray**[2,5], **Kiran Garimella**[3], **Gareth Tyson**[2,4]
[1]The University of Edinburgh, [2]The Alan Turing Institute
[3]Massachusetts Institute of Technology, [4]Queen Mary University of London, [5]The University of Cambridge
steven.wilson@ed.ac.uk, wmagdy@inf.ed.ac.uk, bmcgillivray@turing.ac.uk, garimell@mit.edu, g.tyson@qmul.ac.uk

Abstract

The choice of the corpus on which word embeddings are trained can have a sizable effect on the learned representations, the types of analyses that can be performed with them, and their utility as features for machine learning models. To contribute to the existing sets of pre-trained word embeddings, we introduce and release the first set of word embeddings trained on the content of Urban Dictionary, a crowd-sourced dictionary for slang words and phrases. We show that although these embeddings are trained on fewer total tokens (by at least an order of magnitude compared to most popular pre-trained embeddings), they have high performance across a range of common word embedding evaluations, ranging from semantic similarity to word clustering tasks. Further, for some extrinsic tasks such as sentiment analysis and sarcasm detection where we expect to require some knowledge of colloquial language on social media data, initializing classifiers with the Urban Dictionary Embeddings resulted in improved performance compared to initializing with a range of other well-known, pre-trained embeddings that are order of magnitude larger in size.

Keywords: word embeddings, urban dictionary, slang, sentiment, sarcasm

1. Introduction

Word embeddings provide useful representations of the meanings of words in the form of vectors, and have become the de-facto mapping from tokens to fixed-length continuous inputs for machine learning models in the field of natural language processing. These vectors can be randomly initialized and subsequently learned from the training data for a specific task. However, the ability to pre-train embeddings on a large corpus in an unsupervised way is a powerful technique for transferring generally applicable, word-level semantic information from these massive corpora to a wide range of downstream tasks. Especially in settings with a smaller amount of domain-specific training data, initializing models with these pre-trained embeddings can provide useful representations for words that may not have been seen in the training data and help models converge more quickly during training time. As training time is reduced, the fact that these models can be trained once and subsequently shared with other researchers reduces the environmental impact of having each team of researchers recompute new embeddings, placing the paradigm of training and sharing within the principles of Green AI (Schwartz et al., 2019).

When training word embeddings, there are a number of algorithms and parameters to choose from and tune, but it has also been shown that the choice of *corpus* on which to pre-train is also extremely important (Nooralahzadeh et al., 2018; Risch and Krestel, 2019). This not only affects the vocabulary – and therefore the coverage – of the embeddings, but also shifts the meaning of other words to match the way that they are used in the corpus. For certain cases, such as the modeling and analysis of more descriptive mood and sentiment topics (Fast et al., 2016) or for cases in which the application domain greatly differs from typical large web or news corpora (Silva and Amarathunga, 2019), the choice of pre-trained word embeddings may have an impact on both model performance and the ability to analyze results. That is, embeddings learned from different corpora also provide researchers with the ability to analyze the distributional semantics of the words as they appear in that specific corpus, allowing for comparisons of learned representations across different corpora (Tan et al., 2015).

Several important sets of pre-trained word embeddings have been released and leveraged in multitudinous applications (Mikolov et al., 2013; Pennington et al., 2014; Mikolov et al., 2018). Typically, these have been trained on large news, web, or social media corpora, with the aim of learning representations for a wide range of use cases. In these cases, the embeddings are mostly learned from examples of the words being used in context, rather than text that specifically describes their meaning. In light of this, some have made use of human curated dictionaries with explicit word definitions as a training source for word embeddings (Tissier et al., 2017; Bosc and Vincent, 2018), especially in cases of rare words (Pilehvar and Collier, 2017), which would typically be treated as out-of-vocabulary by embedding models that did not receive these words as input during training.

In this paper, we focus on training a set of word embeddings to specifically capture the one important category of these less common words: slang and colloquialisms. To accomplish this, we introduce and release the first set of word embeddings trained on the entire content of Urban Dictionary[1], which is a crowd-built online English language dictionary. On Urban Dictionary, the moderation of content added to the resource itself is also managed by the crowd, and so definitions range from serious descriptions of slang terms to those that are outright offensive, inappropriate, or incorrect.

We show that although these embeddings are trained on this type of noisy data, containing fewer total tokens (by at least an order of magnitude) compared to most popular pre-trained embeddings, they have comparable, and in some cases better, performance across a range of common word embedding evaluations. In addition, in several extrinsic

[1] https://www.urbandictionary.com/

tasks that we expect to require some knowledge of colloquial language, including sentiment analysis and sarcasm detection, initializing our models with the Urban Dictionary Embeddings resulted in higher accuracy and F1-scores than when using a range of other well-known, pre-trained embeddings. Lastly, we provide some examples of the word associations learned by these embeddings, which provide a glimpse into the types of semantics captured by these embeddings.

We provide two versions of the Urban Dictionary embeddings publicly, which we believe act as a valuable language resource for multiple natural language processing (NLP) tasks, especially those that deal with slang text. Our embeddings can be freely downloaded.[2]

2. Related Work

2.1. Corpora for Word Embeddings

Often with the goal of producing generally applicable word embeddings, many popular pre-trained word embeddings have been fit to large and diverse corpora of text from the web such as the Common Crawl.[3] In other cases, news articles (Mikolov et al., 2013) or encyclopedic text (Pennington et al., 2014) have been used, providing sources of data that are typically more formal, less prone to spelling and grammatical errors, and have high coverage over a range of well-defined topics. With the rise of social media as an important source of text corpora for social analysis and the study of everyday language, several sets of pre-trained embeddings have been released that are specifically tailored for this type of data (Pennington et al., 2014; Godin et al., 2015; Shoemark et al., 2019) leading to improved performance on classification tasks in this domain, and creating new opportunities to analyze the language of social media. Our work in this paper adds another useful set of embeddings that we hope will lead to similarly innovative new results and analyses.

2.2. Embeddings for non-standard tokens and expressions

When training word embeddings, the representation for a specific word is learned from the many contexts in which that word appears. This lead to challenges when building representations for rare words, or words that may not appear at all in corpora that are used for pre-training. A simple solution is to treat extremely rare or unseen words as out-of-vocabulary (OOV) and representing them with a standard OOV vector, such as the average of all vectors for words in the vocabulary. However, previous work has shown that these issues can also be partially addressed by composing the meaning of these words using subword embeddings (Bojanowski et al., 2017), or by learning separate representations for high and low frequency words (Sergienya and Schütze, 2015). Another line of works seeks to expand the coverage of word embedding models by providing them with adequate training data from which to learn valid representations for some of these rarer words, or other tokens like emoji (Eisner et al., 2016). Additionally, recent work has shown that the meaning of non-compositional multiword expressions are, as should be expected, difficult to derive from the set of the phrases' constituent word vectors, even for state-of-the-art systems (Shwartz and Dagan, 2019). This implies that it should be useful to sometimes learn embeddings of these phrases themselves, which has been done by treating short phrases as single words when learning embeddings for them (Mikolov et al., 2013). We take a similar approach in our work, but we use the structure of Urban Dictionary to guide the selection of phrases to be joined together, as discussed further in section 3.2.1.

2.3. Research using Urban Dictionary

Prior work has already focused on the study of Urban Dictionary as a corpus (Nguyen et al., 2018), finding that the platform has shown steady usage since its inception in 1999, and that the definitions capture a mixture of opinions, humor, and true meanings of the defined headwords. The authors also found skewed distributions in terms of the number of definitions per word and votes per entry, which we also verify in Section 3.1.. However, in our work, we do not attempt to address the skewness issues, rather, we take Urban Dictionary "as is" as a corpus of language in order to investigate how well distributional semantic models can learn useful representations from this resource in its raw form.

Aside from the study of the linguistic content of Urban Dictionary, it has also been shown that the information present in Urban Dictionary can be valuable for downstream applications. In an entity linking system, Urban Dictionary was used for query expansion: entity strings looked up in Urban Dictionary, and when present, the tags associated with the entry were used to identify additional query terms, leading to increased accuracy on an entity linking task (Dalton and Dietz, 2013). In another study, researchers trained a sequence-to-sequence model to generate plausible explanations for nonstandard English terms by training on data collected from Urban Dictionary. Urban Dictionary was also used as a resource for determining the approximate date that new terms were used and defined in a study of lexical emergence in Modern English (Grieve et al., 2017).

3. Data and Methodology

3.1. Urban Dictionary as a Corpus

Urban Dictionary (UD) is a crowd-sourced dictionary for (mostly) English-language terms or definitions that are not typically captured by traditional dictionaries. In the best cases, users provide definitions for new and emerging language, while in reality, many entries are a mix of honest definitions ("Stan: a crazy or obsessed fan"), jokes ("Shoes: houses for your feet"), personal messages ("Sam: a really kind and caring person"), and inappropriate or offensive language (Nguyen et al., 2018). Each entry, uploaded by a single user, contains a term, its definition, examples, and tags (Figure 1). Further, those who view the definition have the opportunity to provide other definitions to the entry and/or also provide a vote in the form of a "thumbs-up" or a "thumbs-down", and these votes are recorded and used to rank the possible definitions for a given term when it is looked up in Urban Dictionary. Entries in the Urban Dic-

[2] http://smash.inf.ed.ac.uk/ud-embeddings/

[3] https://commoncrawl.org

Figure 1: Example entry on Urban Dictionary, including the head word (1), definition (2), usage examples (3), tags (4), user and date (5), and upvote and downvote counts (6). Words and phrases in color that are also bold and underlined indicate links to other entries on Urban Dictionary.

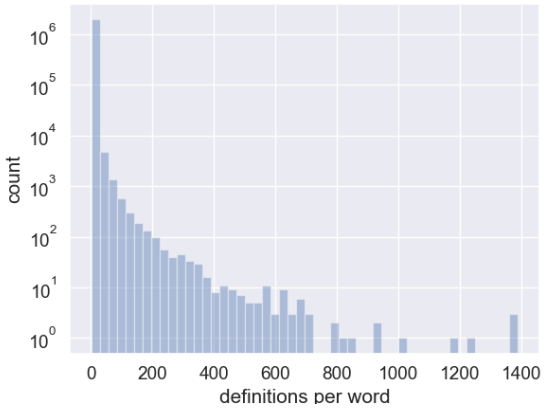

Figure 2: Number of definitions per head word in Urban Dictionary (log scale).

tionary can be for a singular word, a phrase (e.g. "spill the tea", Figure 1), or an abbreviation (e.g. "brb" and "FYI"). For every entry in Urban Dictionary, we crawl and store all of the aforementioned information, resulting in a total of approximately 2 million unique defined terms with an average of 1.8 definitions per term, with the full histogram of the number of definitions per term presented in Figure 2. This data collection includes an up-to-date version of Urban Dictionary as of October 16, 2019. In order to get a high-level understanding of the data, we also visualize the length of each definition (Figure 3) and plot the upvotes and downvotes assigned to the full set of definitions (Figure 4). We note similar skewness in these figures as was reported in earlier analysis of Urban Dictionary data (Nguyen et al., 2018).

3.2. Training Urban Dictionary Embeddings

When training embeddings using standard word embedding approaches, a corpus of running text is required. Therefore, we create a copy of Urban Dictionary in which each entry is represented as a paragraph containing the headword, definition, examples, and tags, each being treated as a separate sentence. This way, the distributional based approaches to learning embeddings will be able to access all elements of an entry when learning representations for the headword,

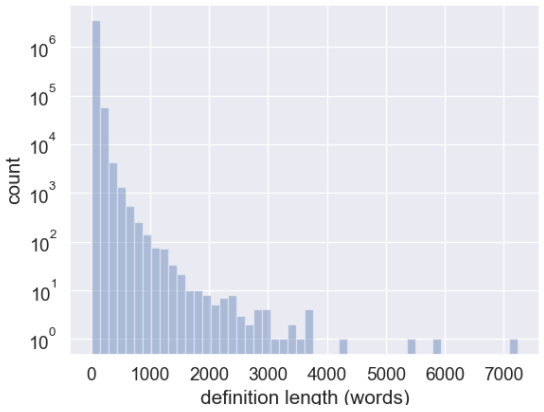

Figure 3: Number of words per definition in Urban Dictionary (log scale).

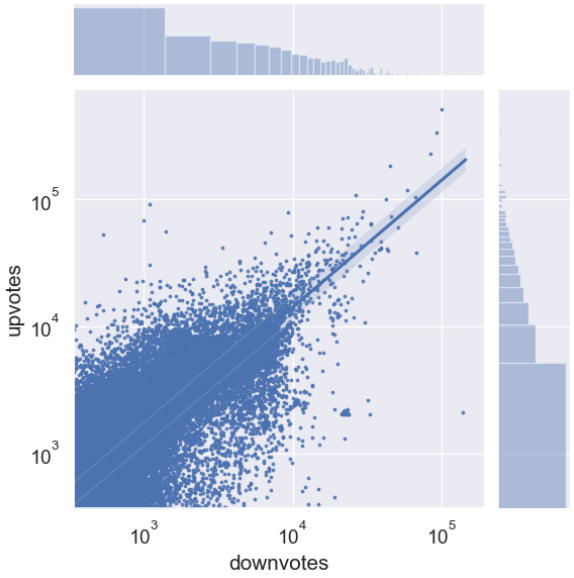

Figure 4: Counts of upvotes and downvotes per entry on Urban Dictionary, with histograms (log scale).

as well as all of the other words that appear in the definitions, examples, and tags. Additionally, we lowercase all text, remove the # character from the tags, and remove punctuation.

We use the fastText framework (Mikolov et al., 2018)[4] in order to train our embeddings on Urban Dictionary. We train our models for 10 epochs using the skipgram architecture and maintain several parameters that were used in the publicly released fastText-CC embeddings: window size of 5, a negative sampling rate of 10, and a word-level dimensionality of 300. We experiment with models that include subword embeddings of 3 to 6 characters and those that do not use subword representations.

3.2.1. Treatment of Multi-word Expressions

Since many of the headwords in Urban Dictionary are actually phrases, we expect that the treatment of multi-word expressions may be important in our training. We experiment with two approaches.

[4] https://fasttext.cc/

In **UD-base**, we tokenize the entire corpus as normal, and provide no explicit guidance about how to join phrases during training time. We do allow the fastText model to learn representations for word n-grams with a length of up to five.
In **UD-phrase**, we join all phrases defined in Urban Dictionary in order to learn embeddings for each of these phrases. To achieve this, we build a list of all multi-word expressions that are used as headwords in Urban Dictionary, and replace all occurrences of these phrases with a single string representing the entire phrase (by replacing whitespace with the underscore character). We make these replacements globally throughout the corpus when training the embeddings, so any occurrence of these phrases in the definitions, examples, or tags will also be joined.
When multiple possible phrases match a specific sequence of words, we greedily choose the longest matching phrase within a given span of text. For example, if "the_best", "friend_ever", and "best_friend_ever" are all possible multi-word expressions, and a document contained the string "she was the best friend ever", the joined version would be "she was the best_friend_ever" since this is the longest of the overlapping phrases (measured in total number of characters).

4. Evaluating the Urban Dictionary Embeddings

We explore the utility of our Urban Dictionary embeddings across a range of intrinsic tasks that directly evaluate the semantics captured by the embeddings. Then, since it has been shown that intrinsic tasks alone to not provide an adequate picture of the utility of word embeddings (Chiu et al., 2016; Faruqui et al., 2016), we test impact of using these embeddings to extract features for several downstream classification tasks that have the potential to benefit from better representations of colloquial language.
We compare a set of pre-trained English language embeddings,[5] including those trained using the word2vec model on the large Google news corpus (Mikolov et al., 2013), two sets of GloVe vectors (Pennington et al., 2014): those trained on web text and those trained on Twitter data, and the publicly released fastText embeddings (Mikolov et al., 2018) trained on web data, which are especially important for comparison given the fact that we also train our embeddings using the fastText library. Details including the size of the training data for each of these sets of pre-trained vectors are presented in Table 1.

	Corpus	Tokens	Vocab size	Dim.
word2vec-gnews	News	100 B	3 million	300
fastText-CC	Web	600 B	2 million	300
GloVe	Web	42 B	1.9 million	300
GloVe-Twitter	Twitter	27 B	1.2 million	200
UD-base	UD	200 M	540,000	300
UD-phrase	UD	200 M	830,000	300

Table 1: Popular word embeddings have typically been pre-trained using tens or hundreds of billions of tokens, whereas Urban Dictionary embeddings were trained on a few hundred million.

4.1. Intrinsic Tasks

First, we evaluate all embeddings across a range of intrinsic evaluation tasks (Jastrzebski et al., 2017) that cover two major categories: semantic similarity, and clustering (Table 2).[6] For the word-level semantic similarity tasks, the goal is to produce a similarity or relatedness score given a pair of tokens. For a set of pairs, the machine generated scores are compared against human-generated gold labels by computing the correlation between the two lists of produced labels. To directly evaluate the ability of the embeddings to capture semantic relatedness, we compute the cosine similarity between the embeddings for each word in the pair, and use this value as the machine generated score. The tasks considered each select the word pairs from different domains, and some have human annotations for slightly different dimensions of semantic similarity. The Wordsim-353 dataset was annotated for *both* strict similarity and general relatedness, allowing us to explore the embeddings' ability to capture each dimension in a controlled setting.
In the clustering tasks, groups of words have been manually sorted based on their semantic properties, and the goal is to recover the original clusters in an unsupervised way using information encoded in the pre-trained word embeddings. For each of these clustering, or categorization tasks, we use both K-means and hierarchical agglomerative clustering methods (Pedregosa et al., 2011) to produce a set of K clusters, where K is the number of categories that exist in the dataset. We then compute the cluster purity scores and report the higher score between the two clustering approaches for each dataset.
While we initially consider word analogy tasks as a third category of intrinsic evaluation, a growing body of recent work has shown that these tests are highly problematic and do not provide a meaningful evaluation of distributional word embedding models (Rogers et al., 2017; Schluter, 2018; Nissim et al., 2019). Therefore, we do not evaluate our embeddings using any analogical reasoning tasks.
Table 3 shows the results of the word-level similarity tasks, and Table 4 shows the results of the word categorization tasks. These results show that although Urban Dictionary is a crowd-sourced resource whose only moderation comes from anonymous volunteers, its content provides word embedding models with representations that capture semantic relationships between words roughly as well as, or better than, popular pre-trained word embedding models, depending on the task considered. This is especially notable in light of the relative size of the training corpora that were used to produce each of these embeddings: the UD embeddings match the performance of embeddings trained on orders of magnitude more data.
The embeddings trained on Urban Dictionary exhibit strong performance on tasks requiring meaningful representations of concrete entities (MEN, BLESS, ESSLI 1a and 2b), nouns (WS353), and ordinary language (RG65), suggesting

[5] A variety of other pre-trained word embeddings were considered, but they consistently underperform the set that we present here on our set of intrinsic tasks.

[6] We used code from the *web* package, located at: `https://github.com/kudkudak/word-embeddings-benchmarks` to run the intrinsic evaluation tasks.

	Task	Domain	Size	Reference
Semantic Similarity	MEN	Image labels	3,000 pairs	(Bruni et al., 2014)
	MTurk	News entities	280 pairs	(Radinsky et al., 2011)
	RG65	Ordinary words	65 pairs	(Rubenstein and Goodenough, 1965)
	RW	Rare words	2,034 pairs	(Luong et al., 2013)
	Simlex999	Word associations	999 pairs	(Hill et al., 2015)
	WS353	Nouns	353 pairs	(Finkelstein et al., 2002)
	WS353-R	Nouns: relatedness	353 pairs	(Finkelstein et al., 2002)
	WS353-S	Nouns: similarity	353 pairs	(Finkelstein et al., 2002)
Clustering	AP	Web nouns	402 words	(Almuhareb and Poesio, 2005)
	BLESS	Concrete nouns	200 words	(Baroni and Lenci, 2011)
	Battig	Category members	5231 words	(Battig and Montague, 1969)
	ESSLI 1a	Concrete nouns	44 words	(Baroni et al., 2008)
	ESSLI 2b	Abstract & concrete	40 words	(Baroni et al., 2008)
	ESSLI 2c	Verbs	45 words	(Baroni et al., 2008)

Table 2: Intrinsic evaluation tasks.

	MEN	MTurk	RG65	RW	SimLex999	WS353	WS353R	WS353S
fastText-CC	0.755	**0.744**	0.790	**0.553**	0.441	0.652	0.611	0.758
GloVe	0.736	0.645	0.817	0.376	0.374	0.553	0.473	0.669
GloVe-Twitter	0.594	0.555	0.698	0.197	0.130	0.451	0.373	0.59
word2vec-gnews	0.741	0.670	0.761	0.471	0.442	0.700	0.635	**0.772**
UD-base	**0.809**	0.697	0.876	0.387	**0.508**	**0.739**	**0.684**	**0.772**
UD-phrase	0.787	0.685	**0.893**	0.393	0.479	0.712	0.656	0.742

Table 3: Performance of pre-trained word embeddings, measured as the correlation between vector similarity scores (computed using cosine similarity between the embedding vectors) and gold standard similarity scores (provided via human annotations), on intrinsic word-level semantic similarity and relatedness tasks.

	AP	BLESS	Battig	ESSLI 1a	ESSLI 2b	ESSLI 2c
fastText-CC	**0.659**	0.755	**0.460**	0.818	0.750	**0.711**
GloVe	0.622	0.785	0.451	0.795	0.750	0.578
GloVe-Twitter	0.515	0.690	0.326	0.773	0.700	0.578
word2vec-gnews	0.649	0.795	0.406	0.750	**0.800**	0.644
UD-base	0.600	0.780	0.389	**0.841**	0.775	0.667
UD-phrase	0.590	**0.800**	0.381	**0.841**	**0.800**	0.622

Table 4: Purity scores achieved using various pre-trained word embeddings on intrinsic word-level clustering tasks.

that the language of Urban Dictionary is relatively concrete in nature. Further, these embeddings showed weaker performances for tasks built on more formal text, like news (MTurk), which is to be expected, since the language of Urban Dictionary is highly informal.

4.2. Extrinsic Tasks

Using the top performing models from our intrinsic evaluations, we train straightforward, but high performing classification models that can showcase the effects of various word embedding initializations. We evaluate these models on two tasks that we expect to benefit from the information present in Urban Dictionary: sentiment analysis and sarcasm prediction in Twitter data. In sentiment analysis, it is important to understand the polarity of terms, and Urban Dictionary provides a useful perspective on slang terms that might not normally be detected as positive (e.g., "lit", "dope", "fire") or negative (e.g., "WOAT", "toolish") using standard approaches. Alternative meanings of words can also be useful to measure in the task of sarcasm detection, where notable incongruity between terms in a text can be an important indicator (Joshi et al., 2017).

4.2.1. Classification Models

For each task, we use the fastText classification model, which is akin to a neural bag-of-words classifier, and has been shown to be extremely competitive with deeper neural architectures including CNN- and RNN-based models (Joulin et al., 2016). We acknowledge that somewhat stronger results would be achieved on these tasks using ensemble methods, using highly task-specific architectures, or by fine-tuning parameter-rich deep learning networks, but the impact of the input embeddings becomes more difficult to observe as model complexity and task-specific tuning increase. Therefore, we choose a model that directly obtains its features from the word embedding dimensions, and learn a simple transformation from these input features to the output space for each task: the embeddings from all words in the input are averaged together, element-wise, and used as a representation for the input text. Then, a logistic regression classifier with a softmax output is trained to make a prediction for the specific task based on this aggregated input.

	w/o word n-grams				*with word n-grams*			
	acc	**prec**	**rec**	**f1**	**acc**	**prec**	**rec**	**f1**
No pre-training	0.596	0.597	0.574	0.571	0.553	0.571	0.506	0.476
fastText-CC	0.596	0.585	0.593	0.585	0.604	0.605	0.584	0.579
GloVe	0.634	0.622	0.632	0.625	0.636	0.624	0.636	0.629
GloVe-Twitter	0.635	0.626	0.631	0.627	**0.650**	**0.642**	**0.644**	**0.642**
word2vec-gnews	0.574	0.565	0.568	0.559	0.581	0.594	0.549	0.537
UD-base	**0.644**	**0.634**	**0.639**	**0.636**	0.634	0.628	0.626	0.626
UD-phrase	0.629	0.620	0.622	0.620	0.640	0.634	0.629	0.629

Table 5: Accuracy, precision, recall, and f1-score achieved on the sentiment prediction task when initializing classifiers with various pre-trained word embeddings.

	w/o word n-grams				*with word n-grams*			
	acc	**prec**	**rec**	**f1**	**acc**	**prec**	**rec**	**f1**
No pre-training	0.790	0.792	0.789	0.789	0.794	0.797	0.793	0.793
fastText-CC	0.788	0.788	0.787	0.787	0.802	0.804	0.802	0.802
GloVe	0.797	0.798	0.797	0.797	0.804	0.805	0.803	0.804
GloVe-Twitter	0.790	0.791	0.790	0.790	0.810	0.811	0.810	0.810
word2vec-gnews	0.776	0.777	0.776	0.776	0.801	0.802	0.800	0.800
UD-base	**0.802**	**0.802**	**0.802**	**0.802**	**0.812**	**0.813**	**0.811**	**0.811**
UD-phrase	0.793	0.793	0.792	0.792	0.802	0.803	0.801	0.801

Table 6: Accuracy, precision, recall, and f1-score achieved on the sarcasm prediction task when initializing classifiers with various pre-trained word embeddings.

Since we expect the meanings of multi-word expressions to be important in our models, we also experiment with using word-level n-grams of up to length 5 for each model, and report these results separately for each task. Though our aim is not necessarily to achieve state-of-the-art results on these tasks, we do still make note of top performing models in the following subsections, order to provide reference points.

4.2.2. Preprocessing

Since all tasks involve the use of Twitter data, we preprocess the input in the same way. We tokenize the text, separate punctuation tokens from alphanumeric tokens, and remove links, user mentions. We also add a copy of each hashtag to the tweet, with the # symbol being removed from the copy and all _ characters in the copy being replace with a single space (i.e, "#great_day" becomes "#great_day great day").

4.2.3. Sentiment Analysis

First, we explore the ability of the Urban Dictionary embeddings as features for sentiment classification on Twitter. We use the SemEval 2017 task A test dataset (Rosenthal et al., 2017), which includes 12,284 English tweets collected by querying for posts related to a range of topics, including named entities, geopolitical entities, and other potentially controversial topics like vegetarianism and gun control. Each tweet has been annotated for its sentiment using a 3-way labeling scheme: positive, negative, and neutral. As allowed by all teams participating in the task, we used training tweets from previous SemEval sentiment analysis tasks for training. Due to tweets that have become unavailable over time, we are able to retrieve approximately 35,000 of the 50,000 annotated tweets provided for training. At the time of the competition, a top scoring model (Cliche, 2017) achieved an accuracy of 0.658 and f1-score of 0.685 (Rosenthal et al., 2017), but other recent work, combining large pre-trained, transformer-based architectures with multi-head attention (Devlin et al., 2019), along with ensemble learning techniques, pushed the f1-score as high as 0.718 (Azzouza et al., 2020).

Table 5 shows the results of our classification models trained using different pre-trained word embeddings, and the results after adding word-level n-grams to the models. We see that in the basic version of the classification model using no word n-grams, initializing with UD-base embeddings consistently gives the best performance, slightly outperforming the GloVe-Twitter embeddings. However, when including the word n-gram features, the GloVe-Twitter embeddings experience a greater boost across all metrics, leading to better performance than any of the Urban Dictionary embeddings. However, the UD-phrase embeddings still outperform all other popular word embeddings as a way to initialize classifier embeddings for the sentiment task.

4.2.4. Sarcasm Detection

Next, observing that sarcastic language is prevalent in Urban Dictionary, we hypothesize that these embeddings will provide a helpful initialization for sarcasm detection models. We train classifiers to predict “sarcastic” or “not sarcastic” given an input tweet. The dataset that we evaluate on for this task (Ptáček et al., 2014) contains examples of tweets that were collected and automatically labeled using a set of sarcasm-indicating hashtags, such as #sarcastic. Using only the text of the tweets, as we do, previous work using intra-attention networks was able to achieve an f1-score as high as 0.860 (Tay et al., 2018). Additionally, it has been shown that an f1-score of 0.934 can be reached by incorporating user-level information into models trained on this dataset (Oprea and Magdy, 2019), which is out of the scope of our current work.

	fastText-CC	GloVe	GloVe-Twitter	word2vec-gnews	UD-base	UD-phrase
great	fantastic	good	good	terrific	good	amazing
	terrific	fantastic	amazing	fantastic	amazing	good
	wonderful	excellent	awesome	trememdous	awesome	the best
	good	wonderful	wonderful	wonderful	best	wonderful
	geat	amazing	fantastic	good	wonderful	awesome
water	water.The	waters	drink	potable water	submergophobia	the water
	Water	salt	ice	Water	monohydrogen	hydrogen oxide
	water.This	drinking	pool	sewage	stormwater	lake michigan highball
	water	dry	bottle	groundwater	waterfuck	hose water
	water.Now	sea	milk	floridian aquifer	butt-splash	stormwater
hogwarts	Hogwarts	hermione	narnia		slitherpuff	harry potter
	Hogwart	dumbledore	potter		potter	slytherin
	slytherin	griffindor	slytherin	n/a	ilvermorny	ilvermorny
	gryffindor	snape	muggle		gryfindor	gryffindor
	hagrid	quidditch	hufflepuff		gryffindor	the harry potter series
yeet	saay	waaaaaaaaaah	yeeet		yeeted	yeet yeet
	ylur	talk-the-talk	thugging		yeeting	yeeted
	daay	pfeh	yuuuuh	n/a	yeet-a-mis-max-mis	yeeticus
	howw	nah-uh	iwu		yote	yeeterday
	buut	megabuckcasinos	werking		yoby	yote
europe	europe.	european	france	european	european	in europe
	germany	germany	germany	germany	embrian	european
	european	asia	european	spain	countries	france
	france	countries	spain	england	germany	germany
	america	france	uk	america	sapmi	baltics
soda	sodas	coke	coke	soda pop	cola	carbonated
	soda-	sodas	drink	sodas	sodagasm	pepsi
	soda.	juice	sprite	soft drink	carbonated	sodie pop
	cola	cola	juice	soft drinks	pepsi	sodagasm
	soda.The	drink	vodka	Lynen ate fruit	mexicoke	sodie
nmhbu	StLafayette				wuzzap	wuzzap
	BlvdMetairie				sappenin'	yagoo
	BlvdMemphis	n/a	n/a	n/a	ishyaboi	yallow ..
	BlvdPhiladelphia				wzup	bookitty
	StreetBaltimore				wwta	best frannn
giraffe	giraffes	elephant	elephant	giraffes	trinko	ostraffe
	hippo	zebra	marius	gorilla	girrhino	hipraffe
	Giraffe	giraffes	zebra	hippopotamus	gipraffe	dementaxcating
	hippopotamus	cheetah	kitten	zebra	mimily	trinko
	okapi	hippo	monkey	rhinoceros	queyon	queyon

Table 7: Query words (left-most column) and their top five nearest neighbors in various pre-trained word embedding models. "n/a" indicates that the query word was not present in the vocabulary of the word embeddings.

Given that Urban Dictionary contains examples of sarcastic language, we expect that the UD-base and UD-phrase word embeddings might provide classifiers with helpful feature for this task. When automatically identifying sarcastic tweets, we find that initializing our classifier with Urban Dictionary embeddings consistently provides better results than using any of the other embeddings, though by a small margin when compared to the GloVe-Twitter initialization (Table 6). This is likely due to the fact that the domain of this prediction task is also Twitter data, making these embeddings especially helpful.

4.2.5. Analysis of Extrinsic Evaluations

Our quantitative results demonstrate the effectiveness of initializing classifiers with word embeddings that have been pre-trained on Urban Dictionary data. The models we use directly leverage the embedding dimensions as features for classification in order to emphasize the effect of initializing with each embedding, and we find comparable or even superior performance when using the Urban Dictionary embeddings for this. This shows that these embeddings are promising for the use in future classification tasks even though they have been trained on a much smaller corpus and have a smaller vocabulary that other popular pre-trained word embeddings.

4.3. Qualitative Analysis

Finally, we take a deeper dive into the types of embeddings that are learned from Urban Dictionary. We select several example terms in order probe the models' ability to capture a variety of word types, and display the top 5 most similar words to each as measured using cosine distance (Table 7). We note that for standard sentiment-related terms, like the example word "great", all models retrieve qualitatively similar groups of positive words. This may explain why Urban Dictionary embeddings perform roughly as well as the other

popular embeddings on the task of sentiment analysis (Section 4.2.3.).

However, when examining the word "water", we immediately start to see a contrast between the different types of embeddings and the associations that are made. Interestingly, the UD-phrase embeddings retrieved some unique ways to describe something as ordinary as water, such as "lake michigan highball" which is defined on Urban Dictionary as "tap water from Chicago". While this is not as standard as related terms like "pool", and "sea" that were found with other models, it provides a unique perspective on the semantics of "water". Here, we also see some inappropriate language among the results, which is another side-effect of training on this type of data, and should be taken into consideration for any applications that produce output based on the full vocabulary space of Urban Dictionary.

Most of the embeddings are able to capture the relationships between words related to the fictional school "hogwarts", yet the Urban Dictionary terms contain what could even be considered nonstandard language relating to this fictional world, such as "slitherpuff", which is only a fan-defined term and not part of the original Harry Potter series of novels (Rowling, 2014).

All of the embeddings appear to have a difficult time capturing the meaning of a word like "yeet" in a way that is not self-referential, though it is worth noting that fastText-CC appears to treat this token as a misspelling of "yet", producing other misspellings as the nearest neighbors. Unsurprisingly, this word doesn't appear at all in the news-based word2vec embeddings.

A crucial example is that of the token "nmhbu", which is short for "not much, how 'bout you?". Most popular embeddings have no representation for this token, and fastText-CC tries to infer the meaning using subword components, which leads it to erroneously match the word to some street name related tokens. Only the Urban Dictionary embeddings are able to find reasonable neighbors for this word, and actually provide a list of other terms that are mostly related to conversations of greeting, which provide a common situation in which "nmhbu" would be used.

On the other hand, the Urban Dictionary embeddings struggle to retrieve useful nearest neighbors for animals like "giraffe", showcasing their lack of encyclopedic knowledge. This lies in contrast to the abilities of the other word embeddings, which consistently identify other animals with similar habitats to a giraffe's. Instead, the Urban Dictionary embeddings give similar words including strange hybrid species like "girrhino" and "ostraffe", which are said to be fusions of giraffes with rhinos and ostriches, respectively. We note that some of these words received very few upvotes (or downvotes) on Urban Dictionary, suggesting that further filtering based on community scoring could help to provide a cleaner set of vocabulary terms in the future.

Overall, we find that the Urban Dictionary embeddings appear to be well suited to capture unique expressions and nonstandard ways of expressing concepts, while sometimes struggling to capture items that require encyclopedic knowledge. When searching through all Urban Dictionary embeddings, it is important to consider that some of the retrieved terms might be offensive or inappropriate, and so care should be taken in cases where these results may be directly presented to unsuspecting users.

5. Conclusions

We have introduced the first set of pre-trained Urban Dictionary word embeddings, which we release as a language resource for analysis and machine learning model initialization. We release both the ud-base and ud-phrase embeddings, as well as a script to automatically perform the multi-word expression-level joining that is needed in order to properly use the ud-phrase embeddings on any corpus. Through a series of intrinsic and extrinsic evaluations, we see that the Urban Dictionary embeddings perform on par with popular and state-of-the-art non-contextual word embeddings. Notably, we observed that the fastText-CC and word2vec-gnews vectors have strong performances on many of the intrinsic tasks, while GloVe and GloVe-Twitter were more beneficial to use as embedding-layer initializations for classification models. For *both* intrinsic and extrinsic tasks, however, the Urban Dictionary Embeddings are on par with, or even outperform, the best of the other popular embedding models, suggesting that they are not overly specialized toward one type of task. Our new embeddings appear to be especially promising when used for tasks involving informal and non-literal language, such as the case of sarcasm detection on social media data. Future work should explore how these embeddings might be incorporated into state-of-the-art models that currently rely on other pre-trained word embeddings, yet are otherwise specifically designed for the tasks for which Urban Dictionary embeddings showed the most potential.

6. Acknowledgements

This work was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1, and EP/S033564/1. We also acknowledge support via EP/T001569/1.

7. Bibliographical References

Almuhareb, A. and Poesio, M. (2005). Concept learning and categorization from the web. In proceedings of the annual meeting of the Cognitive Science society, volume 27.

Azzouza, N., Akli-Astouati, K., and Ibrahim, R. (2020). Twitterbert: Framework for twitter sentiment analysis based on pre-trained language model representations. In Faisal Saeed, et al., editors, Emerging Trends in Intelligent Computing and Informatics, pages 428–437, Cham. Springer International Publishing.

Baroni, M. and Lenci, A. (2011). How we blessed distributional semantic evaluation. In Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics, pages 1–10. Association for Computational Linguistics.

Baroni, M., Evert, S., and Lenci, A. (2008). Lexical semantics: bridging the gap between semantic theory and computational simulation. In Proceedings of the ESSLLI Workshop on Distributional Lexical Semantics 2008. Citeseer.

Battig, W. F. and Montague, W. E. (1969). Category norms of verbal items in 56 categories a replication and extension of the connecticut category norms. *Journal of experimental Psychology*, 80(3p2):1.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bosc, T. and Vincent, P. (2018). Auto-encoding dictionary definitions into consistent word embeddings. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1522–1532.

Bruni, E., Tran, N.-K., and Baroni, M. (2014). Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Chiu, B., Korhonen, A., and Pyysalo, S. (2016). Intrinsic evaluation of word vectors fails to predict extrinsic performance. In Proceedings of the 1st workshop on evaluating vector-space representations for NLP, pages 1–6.

Cliche, M. (2017). BB_twtr at SemEval-2017 task 4: Twitter sentiment analysis with CNNs and LSTMs. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 573–580, Vancouver, Canada, August. Association for Computational Linguistics.

Dalton, J. and Dietz, L. (2013). Umass ciir at tac kbp 2013 entity linking: Query expansion using urban dictionary. In TAC.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186.

Eisner, B., Rocktäschel, T., Augenstein, I., Bosnjak, M., and Riedel, S. (2016). emoji2vec: Learning emoji representations from their description. In Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media, pages 48–54.

Faruqui, M., Tsvetkov, Y., Rastogi, P., and Dyer, C. (2016). Problems with evaluation of word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276*.

Fast, E., Chen, B., and Bernstein, M. S. (2016). Empath: Understanding topic signals in large-scale text. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pages 4647–4657. ACM.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Transactions on information systems*, 20(1):116–131.

Godin, F., Vandersmissen, B., De Neve, W., and Van de Walle, R. (2015). Multimedia lab@ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations. In Proceedings of the workshop on noisy user-generated text, pages 146–153.

Grieve, J., Nini, A., and Guo, D. (2017). Analyzing lexical emergence in modern american english online 1. *English Language & Linguistics*, 21(1):99–127.

Hill, F., Reichart, R., and Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Jastrzebski, S., Leśniak, D., and Czarnecki, W. M. (2017). How to evaluate word embeddings? on importance of data efficiency and simple supervised tasks. *arXiv preprint arXiv:1702.02170*.

Joshi, A., Bhattacharyya, P., and Carman, M. J. (2017). Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)*, 50(5):73.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Luong, T., Socher, R., and Manning, C. (2013). Better word representations with recursive neural networks for morphology. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning, pages 104–113.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., and Joulin, A. (2018). Advances in pre-training distributed word representations. In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018).

Nguyen, D., McGillivray, B., and Yasseri, T. (2018). Emo, love and god: making sense of urban dictionary, a crowd-sourced online dictionary. *Royal Society Open Science*, 5(5).

Nissim, M., van Noord, R., and van der Goot, R. (2019). Fair is better than sensational: Man is to doctor as woman is to doctor. *arXiv preprint arXiv:1905.09866*.

Nooralahzadeh, F., Øvrelid, L., and Lønning, J. T. (2018). Evaluation of domain-specific word embeddings using knowledge resources. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).

Oprea, S. and Magdy, W. (2019). Exploring author context for detecting intended vs perceived sarcasm. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2854–2859, Florence, Italy, July. Association for Computational Linguistics.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543.

Pilehvar, M. T. and Collier, N. (2017). Inducing embed-

dings for rare and unseen words by leveraging lexical resources. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 388–393.

Ptáček, T., Habernal, I., and Hong, J. (2014). Sarcasm detection on czech and english twitter. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 213–223.

Radinsky, K., Agichtein, E., Gabrilovich, E., and Markovitch, S. (2011). A word at a time: computing word relatedness using temporal semantic analysis. In Proceedings of the 20th international conference on World wide web, pages 337–346. ACM.

Risch, J. and Krestel, R. (2019). Domain-specific word embeddings for patent classification. *Data Technologies and Applications*, 53(1):108–122.

Rogers, A., Drozd, A., and Li, B. (2017). The (too many) problems of analogical reasoning with word vectors. In Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017), pages 135–148, Vancouver, Canada, August. Association for Computational Linguistics.

Rosenthal, S., Farra, N., and Nakov, P. (2017). Semeval-2017 task 4: Sentiment analysis in twitter. In Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017), pages 502–518.

Rowling, J. K. (2014). Harry Potter Box Set: The Complete Collection. Bloomsbury Children's Books, 1st edition, October.

Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Schluter, N. (2018). The word analogy testing caveat. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 242–246, New Orleans, Louisiana, June. Association for Computational Linguistics.

Schwartz, R., Dodge, J., Smith, N. A., and Etzioni, O. (2019). Green ai. *arXiv preprint arXiv:1907.10597*.

Sergienya, I. and Schütze, H. (2015). Learning better embeddings for rare words using distributional representations. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 280–285.

Shoemark, P., Liza, F. F., Nguyen, D., Hale, S., and McGillivray, B. (2019). Room to Glo: A systematic comparison of semantic change detection approaches with word embeddings. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 66–76, Hong Kong, China, November. Association for Computational Linguistics.

Shwartz, V. and Dagan, I. (2019). Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, 7:403–419, March.

Silva, A. and Amarathunga, C. (2019). On learning word embeddings from linguistically augmented text corpora. In Proceedings of the 13th International Conference on Computational Semantics - Short Papers, pages 52–58, Gothenburg, Sweden, 23–27 May. Association for Computational Linguistics.

Tan, L., Zhang, H., Clarke, C., and Smucker, M. (2015). Lexical comparison between wikipedia and twitter corpora by using word embeddings. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 657–661.

Tay, Y., Luu, A. T., Hui, S. C., and Su, J. (2018). Reasoning with sarcasm by reading in-between. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1010–1020.

Tissier, J., Gravier, C., and Habrard, A. (2017). Dict2vec: Learning word embeddings using lexical dictionaries. In Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), pages 254–263.